Perspectives in Science (2016) 8, 692—695

ScienceDirect

journal homepage: www.elsevier.com/pisc

Design of a polynomial ring based symmetric homomorphic encryption scheme☆

Smaranika Dasgupta*, **S.K. Pal**

Scientific Analysis Group, DRDO, Delhi, India

Received 17 February 2016; accepted 9 June 2016
Available online 12 July 2016

KEYWORDS
Fully homomorphic encryption;
Cloud computing;
Symmetric FHE;
Polynomial rings;
Refresh

Summary Security of data, especially in clouds, has become immensely essential for present-day applications. Fully homomorphic encryption (FHE) is a great way to secure data which is used and manipulated by untrusted applications or systems. In this paper, we propose a symmetric FHE scheme based on polynomial over ring of integers. This scheme is somewhat homomorphic due to accumulation of noise after few operations, which is made fully homomorphic using a refresh procedure. After certain amount of homomorphic computations, large ciphertexts are refreshed for proper decryption. The hardness of the scheme is based on the difficulty of factorizing large integers. Also, it requires polynomial addition which is computationally cost effective. Experimental results are shown to support our claim.

Introduction

Fully homomorphic encryption (FHE) schemes enable anyone to perform arbitrary homomorphic operations on the ciphertexts which can be finally decrypted to get equivalent results on the plaintexts. Let us consider a fully homomorphic encryption where the encryptions of plaintexts p_1 and p_2 be $Enc(p_1)$ and $Enc(p_2)$ respectively, then both $Enc(p_1+p_2)$ and $Enc(p_1p_2)$ can be computed. FHE is used in cloud computing where processing of encrypted data is required by third parties without the knowledge of secret key. Apart from clouds, there are many more applications of homomorphic encryption schemes such as e-voting, private data processing, encrypted search, etc.

The first fully homomorphic scheme was proposed by Gentry (2009), which involves creating a somewhat homomorphic scheme and further bootstrapping to make it fully homomorphic. Due to computational complexities, Gentry's scheme is practically infeasible. Gentry and Halevi (2011) successfully implemented Gentry's scheme over ideal lattices. Though they incorporated many noticeable optimizations, the scheme still remains computationally complex. The authors van Dijk et al. (2010) presented a simpler scheme over integers instead of ideal lattices, but the key size was too large for practical purposes. Coron et al.

☆ This article belongs to the special issue on Engineering and Material Sciences.

* Corresponding author.
E-mail addresses: smaranikadasgupta@gmail.com (S. Dasgupta), skptech@yahoo.com (S.K. Pal).

http://dx.doi.org/10.1016/j.pisc.2016.06.061

(2011) proposed an idea to reduce the public key size of a homomorphic scheme, which was further reduced by Stehle and Steinfeld (2011). Brakerski and Vaikuntanathan (2011) proposed that a FHE can be based on the hardness of the Learning With Error (LWE) problem. Xiao et al. (2012) showed how hardness of large integer factorization of ciphertext using symmetric key can be the base for security of a homomorphic scheme.

A number of asymmetric FHE schemes have been proposed in the last few years based on ideal lattices, LWE, approximate GCD, etc. For many present day applications, efficient symmetric FHE are also required. Gupta and Sharma (2013) came out with a scheme that used symmetric keys of smaller size based on operations involving matrix computations like matrix inversion, thus making it slightly computationally expensive. Sharma (2014) proposed a scheme which works with single bit and was generalized by Aggarwal et al. (2014) to work on integers. Other symmetric key encryption schemes were proposed by Kipnis and Hibshoosh, Burtyka and Makarevich, Li and Wang.

Mathematical background

Integer factorization problem

Integer factorization is the decomposition of a composite number into its divisors, whose product equals the original integer. Let us take a composite number n. The difficulty in determining its prime factors is referred to as the integer factorization problem.

Ring of polynomials

Let R be a ring. Define $R[x] = \{a_0 + a_1x + \cdots + a_nx^n \mid a_i \in R\}$ where x is indeterminate and $p(x) = a_0 + a_1x + \cdots + a_nx^n$ is called a polynomial in x over R.

Some properties and results:

- $0 + 0.x + \cdots + 0.x^n$ is called the zero polynomial.
- If $p(x) = a_0 + a_1x + \cdots + a_nx^n$; $a_n \neq 0$, $a_i \in R$, then n is called the degree of the polynomial $p(x)$.
- Let $p(x) = \sum_{i=0}^{n} a_ix^i$ and $q(x) = \sum_{i=0}^{m} b_ix^i$ over R of degree n and m respectively. Define '+' on $R[x]$ as $p(x) + q(x) = \sum_{i=0}^{\max\{n,m\}} (a_i + b_i)x^i$; $a_i = 0, \quad \forall i > n$; $b_i = 0, \quad \forall i > m$, define '$*$' on $R[x]$ as $p(x) * q(x) = \sum_{i=0}^{n+m} c_ix^i$; $c_i = \sum_{r+s=i} a_r * b_s$, then $R[x]$ forms a ring with respect to the above binary operations and is called the ring of polynomials over R.
- If R is commutative ring with unity or integral domain, then so is $R[x]$.
- Let $p(x) \in R[x]$. $\alpha \in R$ be a zero of $p(x)$ i.e., $p(\alpha) = 0$, then it is called a zero of the ring.

Ring homomorphism

Definition. *(Ring homomorphism)* If R and S are rings, a function $f: R \to S$ is a ring homomorphism if $\forall r_1, r_2 \in R$.

(a) $f(r_1 + r_2) = f(r_1) + f(r_2)$,
(b) $f(r_1 * r_2) = f(r_1) * f(r_2)$,
(c) $f(1_R) = 1_S$.

Lemma. Let R and S be rings and function $f: R \to S$ be a ring map, then

(a) $f(0) = 0$.
(b) $f(-r) = -f(r), \quad \forall r \in R$.

Proof. (a) $f(0) = f(0 + 0) = f(0) + f(0)$. So, $f(0) = 0$.

By (a), $0 = f(0) = f(r + (-r)) = f(r) + f(-r)$ which implies $f(-r) = -f(r)$.

Definition. *(Kernel of a ring map)* The kernel of a ring map $f: R \to S$ is $kerf = \{r \in R | f(r) = 0\}$.

Definition. *(Image of a ring map)* The image of a ring map $f: R \to S$ is $\text{im} f = \{f(r) | r \in R\}$.

Definition. *(Ring isomorphism)* Let R and S be rings. A ring isomorphism from R to S is a bijective ring homomorphism $f: R \to S$.

Some of these concepts are used in the proposed scheme.

Proposed scheme

The proposed scheme is a symmetric FHE based on polynomial rings over integers. Basically it is a somewhat homomorphic encryption scheme which is made fully homomorphic by a refresh mechanism. If M be an integer which is a multiple of another integer N, then we know that for any x_1, x_2, $x_1 \equiv x_2 \text{mod} N$ remains unaffected if $x_1 \equiv x_2 \text{mod} M$ is performed. This concept is used to construct the refresh key, which is made publicly available without revealing the secret key.

The security parameter of the scheme is ℓ. Message space is $\mathbb{Z}_N$. The integer message is taken in its binary form or the message can be a binary number of several bits. Suppose the binary message (m) be of length $(n+1)$ bits.

Algorithm 1. *KeyGen*(ℓ)

1. Generate secret key S_k, a prime number of length ℓ bits.
2. Choose an even integer z of length γ randomly where $\gamma = \log_2 \ell$.
3. Refresh key $R_k = z * S_k$.

Let m be encoded into message polynomial $m_p(x)$ of degree n with coefficients representing each bit of m.

Algorithm 2. *Enc* $(m_p(x), S_k, n)$

1. Choose a polynomial $y(x)$ of degree n such that $m_p(x) \equiv y(x) \text{mod} S_k$.

2. Randomly pick a polynomial $d(x)$ of degree n.
3. Coefficients of $d(x)$ are integer of length ℓ^a.
4. Compute $c(x) = y(x) + S_k * d(x)$.
5. $c(x)$ is the encrypted message polynomial.

Algorithm 3. Dec $(c(x), S_k, n)$

1. Compute $c(x) \bmod S_k \bmod 2 = m_p(x)$.

After some operations on the ciphertext, the noise produced may exceed thereby making decryption incorrect. In such case, one step refresh procedure is used to eliminate the noise.

Algorithm 4. Refresh $(c(x), R_k, n)$

1. Compute $c'(x) = c(x) \bmod R_k$.

Homomorphic properties

Addition/multiplication of two binary polynomial plaintexts (OR/AND operations) is homomorphic to addition/multiplication of two polynomial ciphertexts with integer coefficients.

Example: Let $S_k = 13$

$m_1 = 65 \equiv 1000001$

$m_{p_1}(x) = x^6 + 1$

$y_1(x) = 27x^6 + 13x^5 + 26x^3 + 14$

$d_1(x) = 2650x^6 + 995x^5 + 259x^2 + 100$

$c_1(x) = 34477x^6 + 12948x^5 + 26x^3 + 3367x^2 + 1314$

$c_1(x) \bmod S_k \bmod 2 = x^6 + 1 \equiv m_{p_1}(x)$

$m_2 = 56 \equiv 111000$

$m_{p_2}(x) = x^5 + x^4 + x^3$

$y_2(x) = 14x^5 + 27x^4 + 40x^3$

$d_2(x) = 119x^5 + 224x^4 + 17x^3 + 2249x^2 + 36$

$c_2(x) = 1561x^5 + 2939x^4 + 261x^3 + 29237x^2 + 468$

$c_2(x) \bmod S_k \bmod 2 = x^5 + x^4 + x^3 \equiv m_{p_2}(x)$

• *Additive homomorphic*

$[(c_1 + c_2)(x)] \bmod S_k \bmod 2 = x^6 + x^5 + x^4 + x^3 + 1$
$\equiv (m_{p_1} + m_{p_2})(x)$

$m_1 + m_2 = 1111001$

• *Multiplicative homomorphic*

$[(c_1 * c_2)(x)] \bmod S_k \bmod 2 = x^{11} + x^{10} + x^9 + x^5 + x^4 + x^3$
$\equiv (m_{p_1} * m_{p_2})(x)$

$m_1 * m_2 = 111000111000$

Observations and analysis

In this scheme the length of polynomial $d(x)$ is taken ℓ^a bits. For security and computational efficiency, the value of a has been taken as 3, 4, 5, 6. The proposed scheme was implemented in C using GNU multi precision library (GMP) on 3 GHz processor with 4 GB RAM. Taking coefficient of $d(x)$ from $\mathbb{Z}$ of length ℓ^4 bits, observed runtimes of encryption and decryption for 5 and 6 bits of message are given in Table 1. The graph shown in Fig. 1 depicts security parameter length versus encryption and decryption time in seconds for 5 and 6 bits of message with coefficient of $d(x)$ from $\mathbb{Z}$ of length ℓ^5 bits. For all the given bit lengths, key generation time was in fraction of milliseconds.

Table 1 Runtimes of encryption and decryption modules.

Bits	5 bits of message		6 bits of message	
	Encryption (ms)	Decryption (ms)	Encryption (ms)	Decryption (ms)
64	20	13	20	18
80	45	66	48	78
96	84	140	100	167
128	255	493	306	578
192	1477	3364	2247	6409

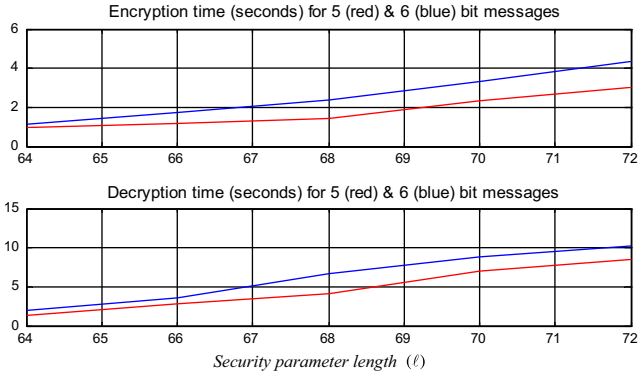

Figure 1 Security parameter length versus encryption and decryption time.

Conclusions and future work

In this paper we proposed a symmetric FHE scheme based on polynomial rings over integers. Experimental results show that the scheme can be used for present day practical applications. Future work will include cryptanalysis of symmetric key FHE schemes and design of practically usable ring based asymmetric FHE schemes.

References

Aggarwal, N., Gupta, C.P., Sharma, I., 2014. Fully homomorphic encryption scheme without bootstrapping. In: CCIOT 2014, pp. 14—16.

Brakerski, Z., Vaikuntanathan, V., 2011. Efficient fully homomorphic encryption from (standard) LWE. In: FOCS.

Coron, J.-S., Mandal, A., Naccache, D., Tibouchi, M., 2011. Fully homomorphic encryption over the integers with shorter public-keys. In: Advances in Cryptology-CRYPTO 2011, Springer, Berlin, Heidelberg, pp. 487—504.

Gentry, C., 2009. A fully homomorphic encryption using ideal lattices. In: STOC, vol. 9, pp. 169—178.

Gentry, C., Halevi, S., 2011. Implementing Gentry's fully homomorphic encryption scheme. In: EUROCRYPT 2011, LNCS. Springer.

Gupta, C.P., Sharma, I., 2013. A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds. In: NOF 2013, pp. 23—25.

Sharma, I., 2014. A symmetric FHE scheme based on Linear Algebra. In: IJCSET, vol. 5.

Stehle, D., Steinfeld, R., 2011. Faster fully homomorphic encryption, Cryptology ePrint archive, Report.

van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V., 2010. Fully homomorphic encryption over the integers. In: Advances in Cryptology-EUROCRYPT 2010. Springer, pp. 24—43.

Xiao, L., Bastani, O., I-Ling, Y., 2012. An efficient homomorphic encryption protocol for multi-user systems. IACR Cryptology ePrint Archive, 193.